

Curso: - Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
- Superior de Tecnologia em Redes de Computadores

Disciplina: - Algoritmos e Técnicas de Programação
- Introdução à Programação

Professor: Fabiano Papaiz

ESTRUTURA DE REPETIÇÃO FOR (PARA)

Em C, a estrutura de repetição PARA é escrita através da instrução FOR.

Possui a seguinte sintaxe:

```
for ( inicialização; condicao_de_execucao; incremento/decremento ) {  
    //codigo que será executado enquanto condição_de_execucao for TRUE  
}
```

A seguir está um exemplo de utilização do FOR. O programa irá exibir na saída padrão todos os números de 1 a 100:

```
for ( int i = 1; i <= 10; i++) {  
    //imprime na saída o valor atual da variável "i"  
    cout << i << endl;  
}
```

Percebemos então, que o FOR possui três seções que são separadas através do ; (ponto-e-vírgula). São as seções de:

- 1ª. inicialização,
- 2ª. condição de execução e
- 3ª. incremento/decremento.

Na primeira seção do FOR, nós inicializamos a variável de controle para execução do FOR. No exemplo, utilizamos uma variável do tipo int chamada i e atribuímos o valor 1 para ela: **int i = 1.**

Na segunda seção, definimos qual a condição lógica que determinará até quando o FOR será executado: **i <= 10**. No exemplo, o FOR será executado enquanto o valor de i for menor ou igual a 10.

Na terceira seção, definimos como a variável de controle será alterada a cada iteração (repetição) do FOR. Geralmente incrementamos ou decrementamos o valor da variável: **i++**. Neste exemplo, significa que **após cada execução do código dentro do FOR** a variável i terá seu valor incrementado em 1, ou seja, na 1ª execução o valor de i será 1, na 2ª execução será 2 e assim por diante. Note que em C, **i++** é o mesmo que **i = i + 1**. Da mesma forma, **i--** é o mesmo que **i = i - 1**.

Exercícios utilizando a instrução FOR:

1. Crie um programa que escreva que exiba os números pares no intervalo de 1 a 100.
2. Crie um novo programa baseado no exercício 1, de forma que, após a exibição dos números pares, sejam também exibidos os números ímpares de 1 a 250.
3. Crie um programa que exiba a seguinte saída:

```
2
4
8
16
32
```

4. Crie um programa que contenha o código abaixo:

```
for (int i = ?; i <= ?; i++) {
    for (int j = ?; j <= ?; j++) {
        cout << "[" << ? << "." << ? << " "; //isto irá exibir "[N.M] "
    }
    cout << endl;
}
```

Altere neste código os locais marcados com ? de forma que o programa exiba a seguinte saída:

```
[1.200] [1.201] [1.202] [1.203] [1.204] [1.205] [1.206] [1.207] [1.208] [1.209] [1.210]
[2.200] [2.201] [2.202] [2.203] [2.204] [2.205] [2.206] [2.207] [2.208] [2.209] [2.210]
[3.200] [3.201] [3.202] [3.203] [3.204] [3.205] [3.206] [3.207] [3.208] [3.209] [3.210]
[4.200] [4.201] [4.202] [4.203] [4.204] [4.205] [4.206] [4.207] [4.208] [4.209] [4.210]
[5.200] [5.201] [5.202] [5.203] [5.204] [5.205] [5.206] [5.207] [5.208] [5.209] [5.210]
```